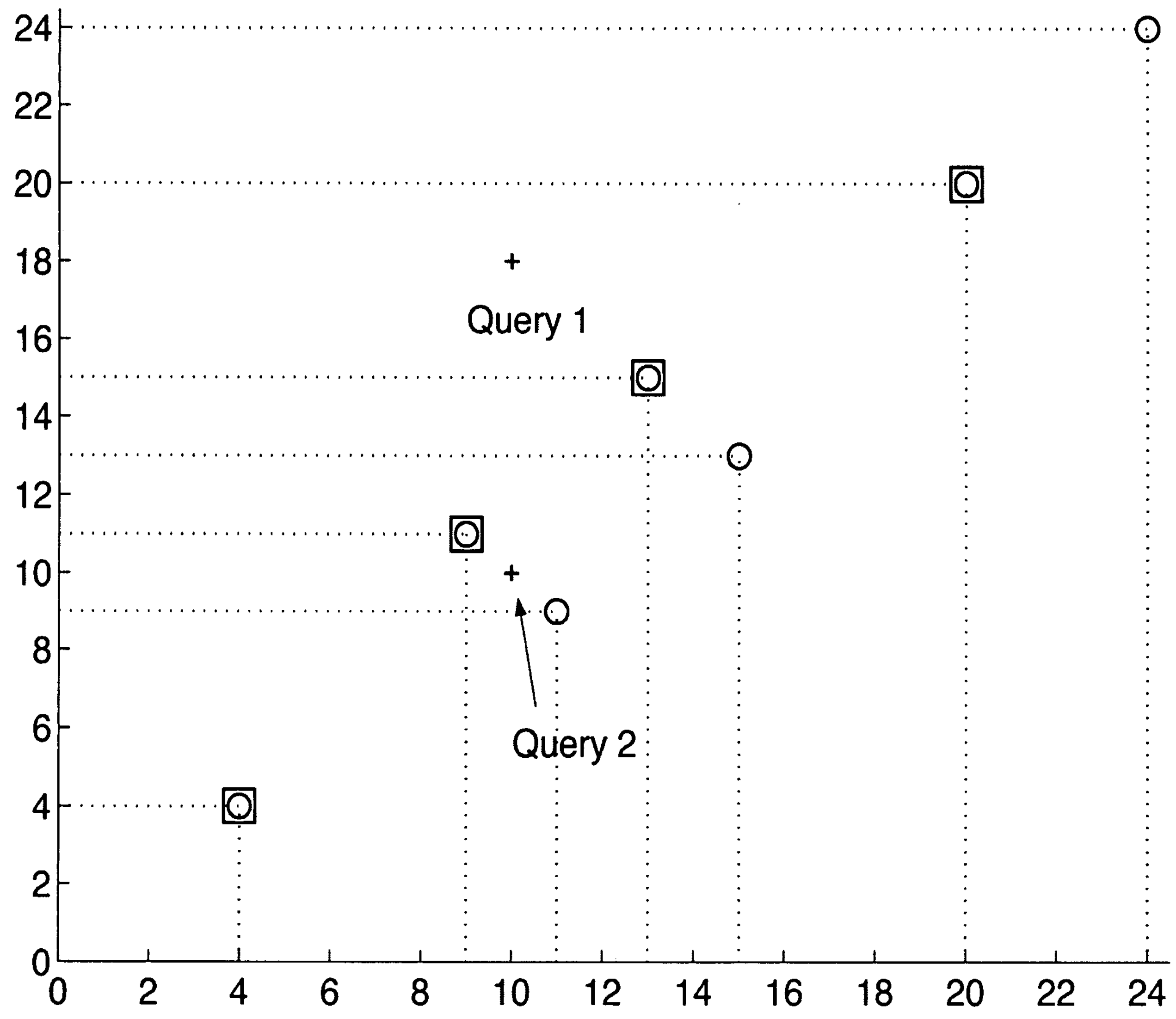

24
22
20
18
16
14
12
10
8
6
4
2
0
0
2
4
6
8
10
12
14
16
18
20
22
24
Query 1
Query 2

FIG. 1

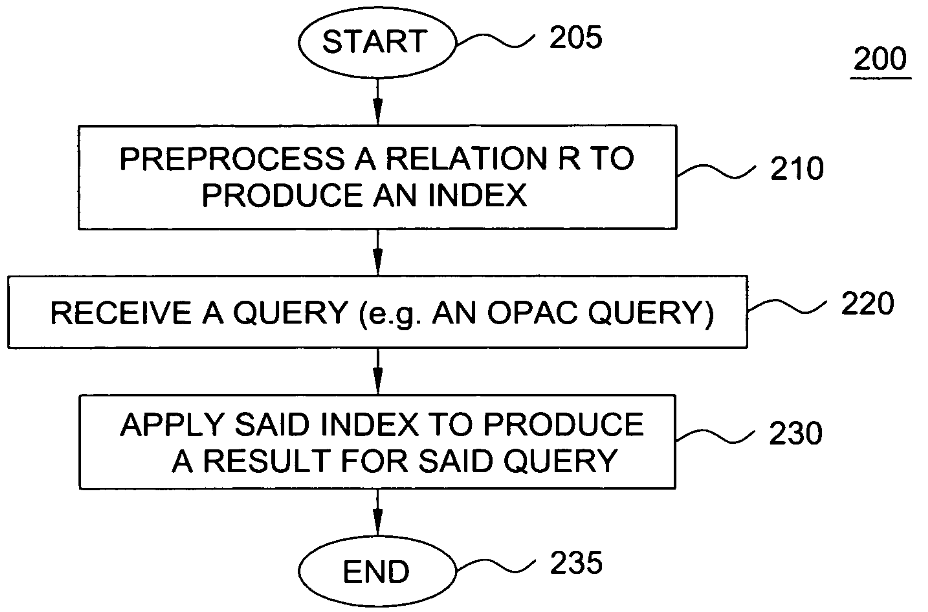

START
205
200
PREPROCESS A RELATION R TO PRODUCE AN INDEX
210
RECEIVE A QUERY (e.g. AN OPAC QUERY)
220
APPLY SAID INDEX TO PRODUCE A RESULT FOR SAID QUERY
230
END
235

FIG. 2

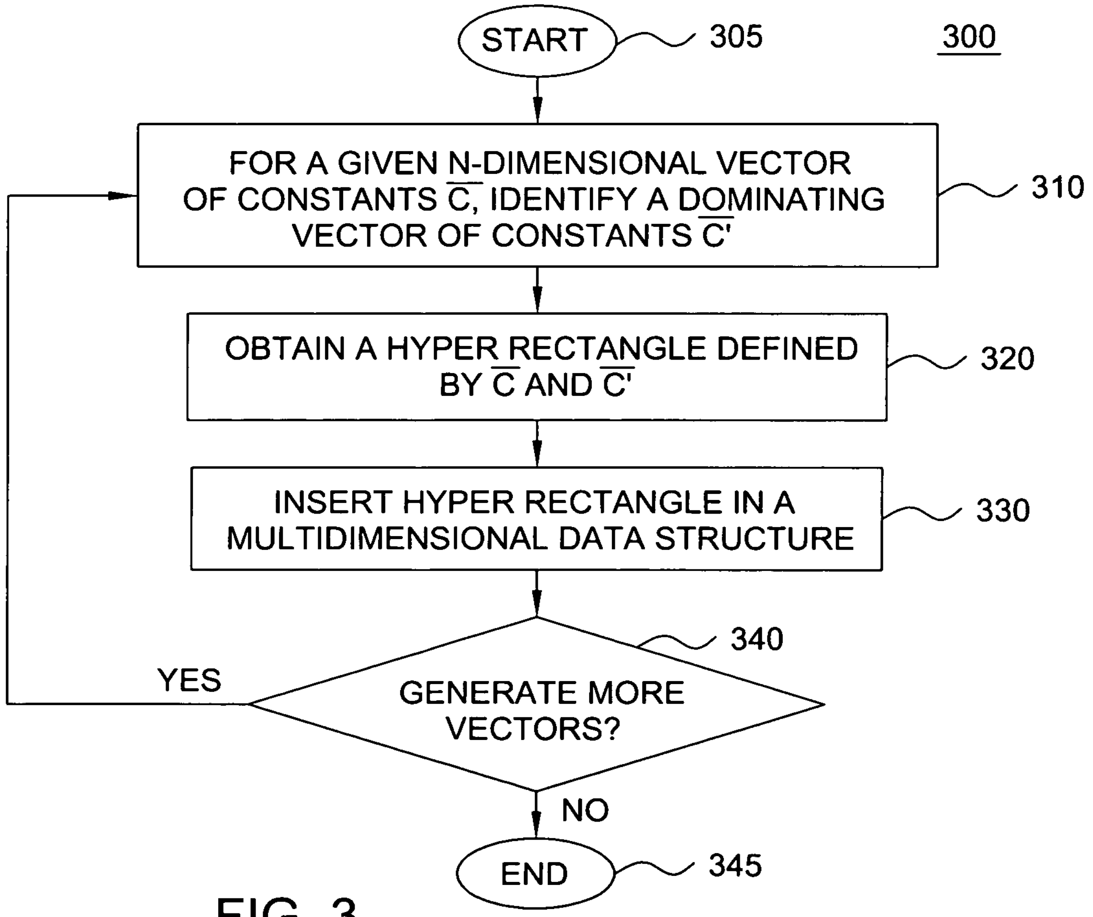

START
305
300
FOR A GIVEN N-DIMENSIONAL VECTOR OF CONSTANTS $\overline{C}$, IDENTIFY A DOMINATING VECTOR OF CONSTANTS $\overline{C'}$
310
OBTAIN A HYPER RECTANGLE DEFINED BY $\overline{C}$ AND $\overline{C'}$
320
INSERT HYPER RECTANGLE IN A MULTIDIMENSIONAL DATA STRUCTURE
330
GENERATE MORE VECTORS?
340
YES
NO
END
345

FIG. 3

```
Algorithm **GeneratePartitions**($\epsilon, \epsilon'$, $D$)

 Initialize:
    Q: Queue of multidimensional constraint vectors
    $\mathcal{R}$: R-tree
    $s, c, c'$: constraint vectors
       each coordinate of $s$ is initially set to be
       equal to $D$ and, and $s$ is added to Q

while Q not empty
   $\bar{c}$ = headof(Q)
   $(r, \bar{C}, p, S)$ = LocateSolution($\bar{c}$)
   if there is no rectangle $r'$ in the R-tree $\mathcal{R}$
    that contains rectangle $r$ and $r$ not NULL
      Insert $(r, p, \bar{C}, S)$ to the R-tree $\mathcal{R}$
      by storing $(r, p, \bar{C})$ in a leaf index entry
      and maintaining a pointer to the set of
      tuple identifiers in the solution $S$ on disk
      CreateFront(Q,r)
   endif
end-while


Algorithm **LocateSolution**($\bar{c}$)

**Input:** constant vector $\bar{c} = (c_1, \ldots c_n)$
**Output:** $(r, \bar{C}, p.S)$
(p,S) = MultiKnapsack($\bar{c}$)
if (S is NULL) return (NULL, NULL, 0, NULL)
for i = 1 to n
   $c_i' = \frac{c_i}{1+\epsilon}$
$(p', S')$ = MultiKnapsack($\bar{c}'$)
if ($S'$ is NULL) return (NULL, NULL, 0, NULL)
if ( $(1+\epsilon')p' > p$ )
   while ($p' \geq \frac{p}{1+\epsilon'}$)
      $\bar{c}_t = \bar{c}'$; $p_t = p'$; $S_t = S'$
      for i = 1 to n
         $c_i' = \frac{c_i'}{1+\epsilon'}$
      $(p', S')$ = MultiKnapsack($\bar{c}'$)
   end-while
   return (FormRect($\bar{c}_t$,$\bar{c}$), $\bar{c}_t$, $p_t$, $S_t$)
else
   return (FormRect($\bar{c}'$, $\bar{c}$), $\bar{c}$, p, S)
```

FIG. 4

DATABASE

510

	c1	c2	Profit
t1	10	5	100
t2	5	10	100

RESULT FOR c1 & c2 <=15

c1	c2	P	Solution
5	10	100	(t2)
10	5	100	(t1)
15	15	200	(t1, t2)

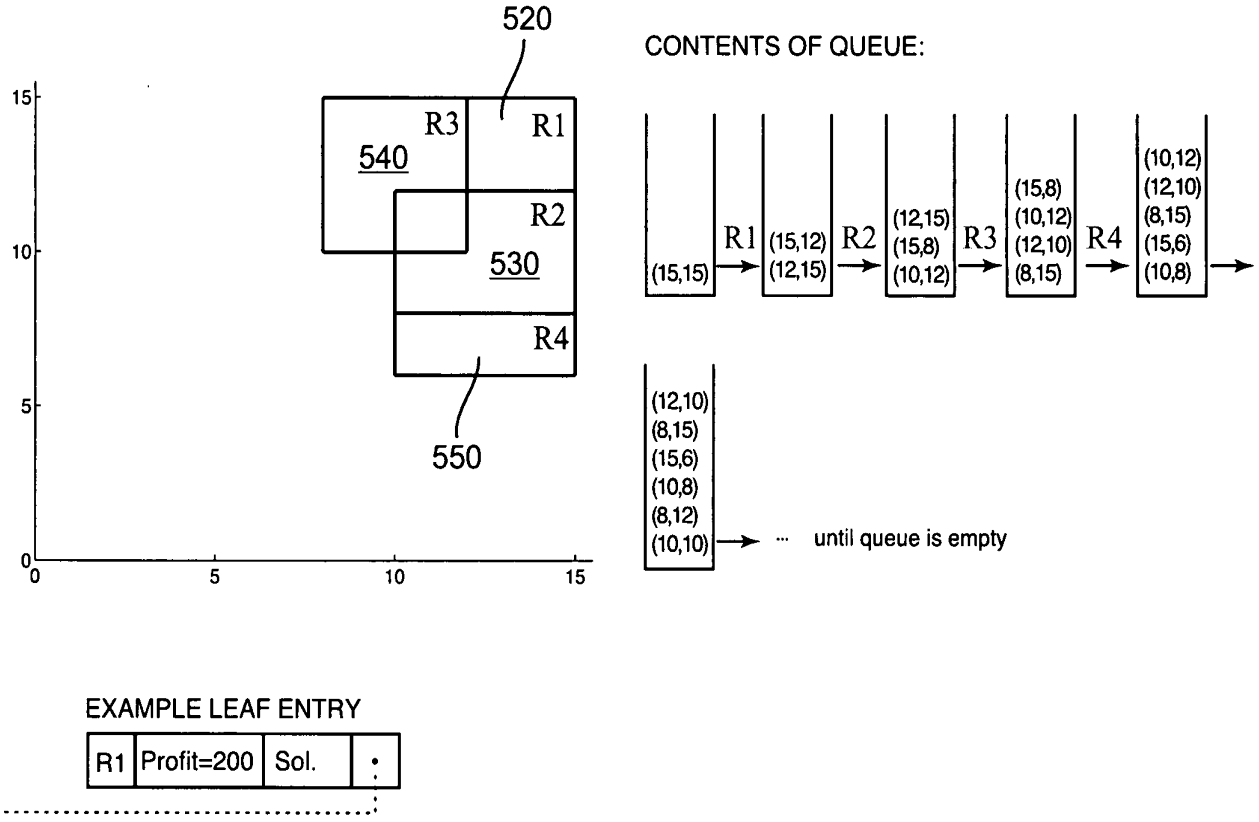


FIG. 5

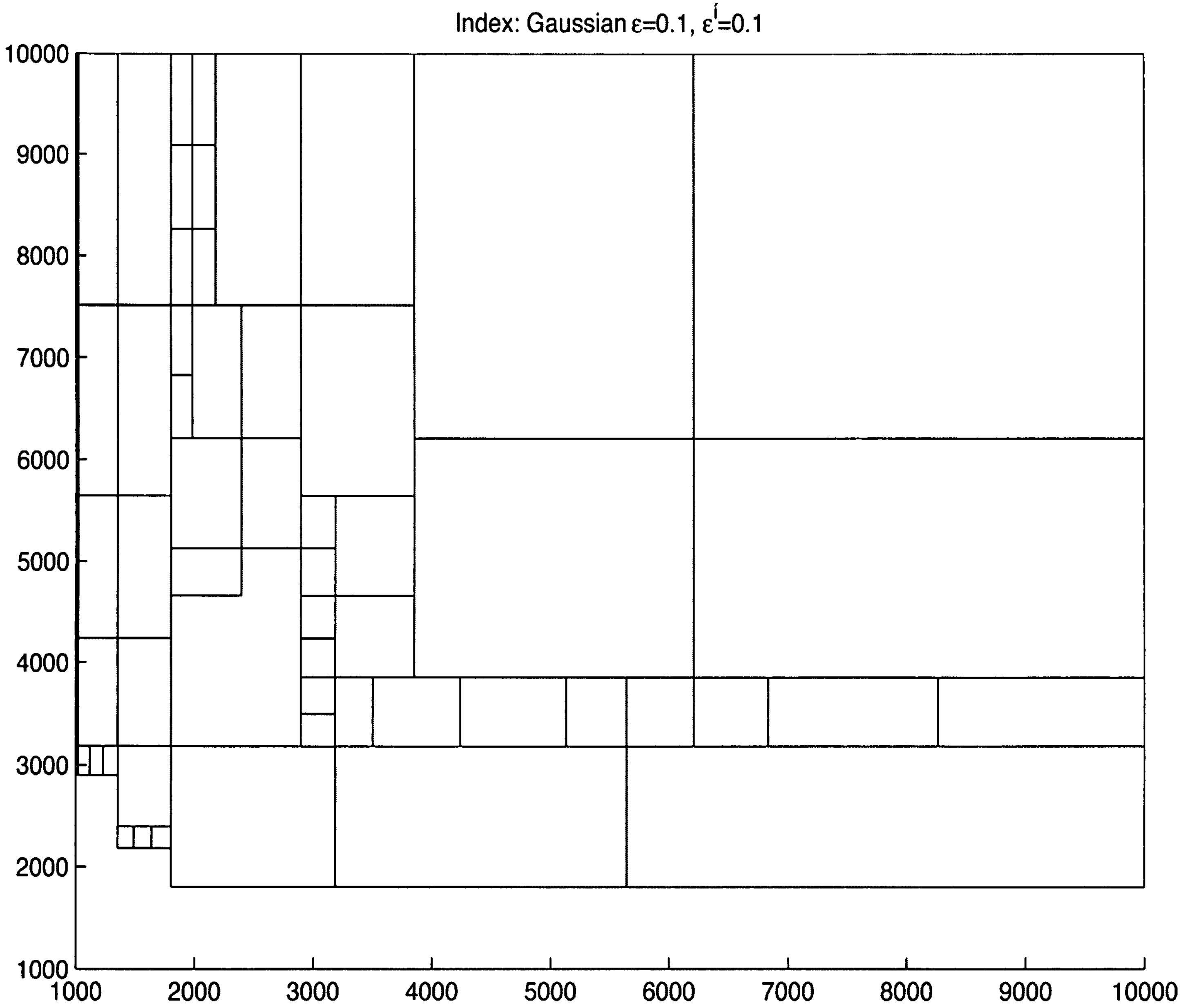
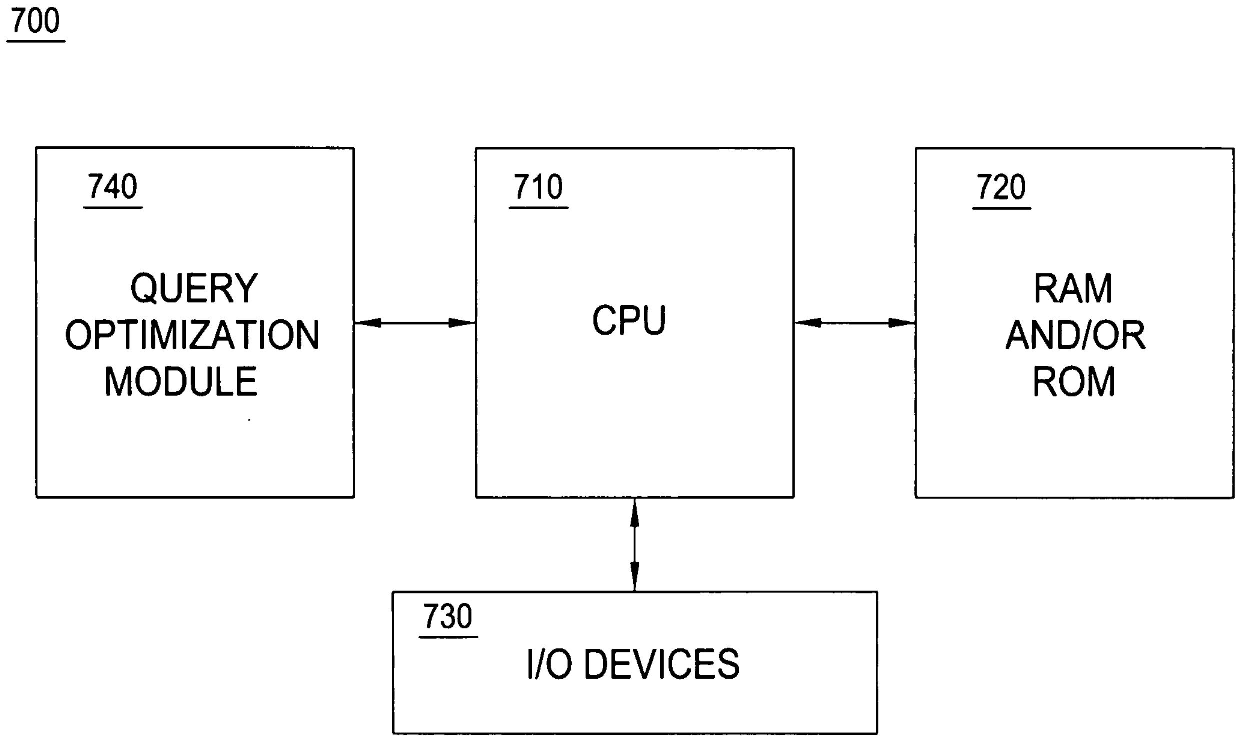

Index: Gaussian ε=0.1, ε'=0.1
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

FIG. 6

FIG. 7


700
740
QUERY
OPTIMIZATION
MODULE
710
CPU
720
RAM
AND/OR
ROM
730
I/O DEVICES